080398.P581 PATENT

UNITED STATES PATENT APPLICATION

for

SEMANTICS-BASED MOTION ESTIMATION FOR MULTI-VIEW VIDEO CODING

Applicants:

Sundar Vedula
Rohit Puri
Ali J. Tabatabai

prepared by:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN
12400 Wilshire Boulevard
Los Angeles, CA 90026-1026
(408) 720-8598

EXPRESS MAIL CERTIFICATE OF MAILING

"Express Mail" mailing label number EV409361121US
Date of Deposit March 31, 2004
I hereby certify that this paper or fee is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231.
Michelle Begay
(Typed or printed name of person mailing paper or fee)
Michelle Begay
(Signature of person mailing paper or fee)

SEMANTICS-BASED MOTION ESTIMATION FOR MULTI-VIEW VIDEO CODING

RELATED APPLICATIONS

[0001] The present application claims priority to U.S. Provisional Application Serial No. 60/493,883, filed August 7, 2003, which is incorporated herein in its entirety.

FIELD OF THE INVENTION

[0002] The invention relates to video coding in general. More particularly, the invention relates to performing motion estimation for multi-view video coding.

COPYRIGHT NOTICE/PERMISSION

[0003] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawings hereto: Copyright © 2004, Sony Electronics, Inc., All Rights Reserved.

BACKGROUND OF THE INVENTION

[0004] Motion estimation and compensation has proven to be an effective method to reduce the overall bit rate of video sequences. Motion estimation is a process for estimating the motion of image samples (e.g., pixels)

between frames. Using motion estimation, the encoder attempts to match blocks of pixels in one frame with corresponding pixels in another frame. After the most similar block is found in a given search area, the change in position of the corresponding pixels is approximated and represented as motion data, such as a motion vector. Motion compensation is a process for determining a predicted image and computing the error between the predicted image and the original image. Using motion compensation, the encoder applies the motion data to an image and computes a predicted image. The difference between the predicted image and the input image is called the error signal.

[0005] Conventional motion estimation and compensation methods have been used by various encoders (e.g., MPEG-x encoders, H.26x encoders, etc.), enabling efficient cross-time compression of single-view video sequences. However, while matches produced by these methods may be efficient from a compression perspective, they are often semantically incorrect because they need not represent the underlying "true" motion in the video sequence.

SUMMARY OF THE INVENTION

[0006] A motion estimation method and apparatus for video coding of a multi-view sequence is described. An exemplary motion estimation method includes identifying one or more pixels in a first frame of a multi-view video sequence, and constraining a search range associated with a second frame of the multi-view video sequence based on an indication of a desired correlation

between efficient coding and semantic accuracy. The semantic accuracy relies on use of geometric configurations of cameras capturing the multi-view video sequence. The method further includes searching the second image within the constrained search range for a match of the pixels identified in the first frame.

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] The present invention will be understood more fully from the detailed description given below and from the accompanying drawings of various embodiments of the invention, which, however, should not be taken to limit the invention to the specific embodiments, but are for explanation and understanding only.

[0008] **Figure 1** is a block diagram of one embodiment of an encoding system.

[0009] **Figure 2** is a flow diagram of one embodiment of a process for performing motion estimation for a multi-view video sequence.

[0010] **Figure 3** is a flow diagram of one embodiment of a process to define a search range for motion estimation using a seeding approach.

[0011] **Figure 4** illustrates two exemplary frames of a multi-view video sequence.

[0012] **Figure 5** illustrates a comparison of bitrates obtained by experimenting with different motion estimation methods for a multi-view video sequence.

[0013] **Figure 6** is a block diagram of a computer environment suitable for practicing embodiments of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

[0014] In the following detailed description of embodiments of the invention, reference is made to the accompanying drawings in which like references indicate similar elements, and in which is shown, by way of illustration, specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical, functional and other changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

[0015] Beginning with an overview of the operation of the invention, **Figure 1** illustrates one embodiment of an encoding system 100. The encoding system 100 performs multi-view coding in accordance with video coding standards such as MPEG-x and H-26x. The encoding system 100 may be implemented in hardware, software, or a combination of both. In software implementations, the encoding system 100 may be stored and distributed on a variety of conventional computer readable media. In hardware implementations, the modules of the encoding system 100 are implemented in digital logic (e.g., in

an integrated circuit). Some of the functions can be optimized in special-purpose digital logic devices in a computer peripheral to off-load the processing burden from a host computer.

[0016] The encoding system 100 includes a signal receiver 102, a motion estimator 104, a motion compensator 106, a coding module 108, a buffer 110, and a frame reconstructor 112. The frame receiver 102 is responsible for receiving a video signal with a multi-view video sequence and forwarding individual frames contained in the multi-view video sequence to the motion estimator 104.

[0017] The motion estimator 104 is responsible for comparing a current frame of the multi-view video sequence with a frame reconstructed from a previous frame and stored in the buffer 110, and estimating motion in the current frame with respect to the previous frame. In particular, the motion estimator 104 searches the reconstructed previous frame for a match of each pixel (or a block of pixels) from the current frame to compute a motion vector for each pixel or block. The resulting motion vectors are passed on to an output of the encoding system 100.

[0018] The motion compensator 106 is responsible for reading the motion vectors and the reconstructed previous frame, computing a predicted image for the current frame, and subtracting the predicted image from the current frame, which results in a residual frame.

[0019] The coding module 108 is responsible for subjecting the residual frame to various encoding operations to compress the signal, and passing the compressed signal to the output of the encoding system 100. Examples of the encoding operations may include, for example, Discrete Cosine Transform (DCT) in combination with adaptive quantization, differential coding, run-length coding (RLC), variable-length coding (VLC), etc.

[0020] The frame reconstructor 112 is responsible for adding the residual frame to the predicted image to obtain a reconstructed current frame, and storing the reconstructed current frame in the buffer 110 for further use by the motion estimator 104 and the motion compensator 106.

[0021] In one embodiment, the motion estimator 104 includes a block identifier 114, a search range determinator 116, a searcher 118, and a motion vector calculator 120. The block identifier 114 is responsible for receiving a frame from a multi-view video sequence from the signal receiver 102, dividing this current frame into blocks (or individual pixels), and passing each block (or pixel) to the search range determinator 116.

[0022] The search range determinator 116 is responsible for defining a search range within a previous frame of the multi-view video sequence being coded to perform a search for a matching block. The search range determinator 116 determines the search range based on geometric configurations of cameras used to capture the multi-view sequence. The geometric configurations of cameras define multi-view geometry constraints which, when used to determine

the search range, enable higher semantic accuracy of matches and help reduce complexity of the search.

[0023] In one embodiment, the multi-view geometry constraints are described by the epipolar geometry. In particular, according to the epipolar geometry, for a pair of views looking at the same scene, true semantic matches for any pixel in the first view lie along the epipolar line corresponding to that pixel in the second view. Hence, the search determinator 116 uses the position of the epipolar line in the previous frame to determine the search range for a relevant block. More specifically, the search determinator 116 determines how to constrain the search range with respect to the position of the epipolar line based on a desired correlation between efficient coding and semantic accuracy of matches. That is, for higher semantic accuracy, the search determinator 116 constrains the search range to be closer to the epipolar line. For higher coding efficiency, the search determinator 116 defines the search range to cover a larger area around the epipolar line. In one embodiment, the desired correlation between efficient coding and semantic accuracy of matches is specified by the user (e.g., based on the needs of an application for which the encoding system 100 is used) and can be modified at any point in time.

[0024] The searcher 118 is responsible for searching the previous frame within the determined search range for a matching block, and the motion vector calculator 120 is responsible for computing a motion vector for this block.

[0025] Accordingly, the encoding system 100 combines multi-view geometry constraints with block-based matching to achieve better semantic accuracy of matching. This approach is referred to herein as semantics-based motion estimation. Furthermore, the encoding system 100 can vary the hardness of the constraint to allow a user to control the correlation between efficient compression and semantic accuracy of matching.

[0026] **Figures 2 and 3** are flow diagrams of motion estimation processes that may be performed by a motion estimator 104 of **Figure 1**, according to various embodiments of the present invention. The process may be performed by processing logic that may comprise hardware (e.g., circuitry, dedicated logic, etc.), software (such as run on a general purpose computer system or a dedicated machine), or a combination of both. For software-implemented processes, the description of a flow diagram enables one skilled in the art to develop such programs including instructions to carry out the processes on suitably configured computers (the processor of the computer executing the instructions from computer-readable media, including memory). The computer-executable instructions may be written in a computer programming language or may be embodied in firmware logic. If written in a programming language conforming to a recognized standard, such instructions can be executed on a variety of hardware platforms and for interface to a variety of operating systems. In addition, the embodiments of the present invention are not described with reference to any particular programming language. It will be

appreciated that a variety of programming languages may be used to implement the teachings described herein. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application, module, logic, etc.), as taking an action or causing a result. Such expressions are merely a shorthand way of saying that execution of the software by a computer causes the processor of the computer to perform an action or produce a result. It will be appreciated that more or fewer operations may be incorporated into the processes illustrated in **Figures 2 and 3** without departing from the scope of the invention and that no particular order is implied by the arrangement of blocks shown and described herein.

[0027] **Figure 2** is a flow diagram of one embodiment of a process 200 for performing motion estimation for a multi-view video sequence.

[0028] Referring to **Figure 2**, processing logic begins with identifying one or more pixels in a first frame of the multi-view video sequence (processing block 202). The first frame is a frame being currently processed. The one or more pixels may represent a block or an individual pixel.

[0029] At processing block 204, processing logic accesses a second frame of the multi-view video sequence and computes the epipolar line corresponding to the above pixels of the first frame within the second frame. The second frame is a previously processed frame (e.g., reconstructed by the motion estimator 104 and stored in the buffer 110). In one embodiment, in which motion estimation is performed for individual pixels of the first frame, the epipolar line

corresponds to a relevant pixel. In another embodiment, in which motion estimation is performed for blocks within the first frame, the epipolar line corresponds to a pixel at a predefined position within a relevant block. In one embodiment, the epipolar line is computed using the fundamental matrix (known as F matrix), which is an algebraic representation of the epipolar geometry. The computation of the epipolar line may be expressed as follows:

$$l_1 = Fx_0,$$

wherein l_1 is the epipolar line in the second image that corresponds to the pixel x_0 in the first image.

[0030] It will be understood by a person skilled in the art that the epipolar line can be computed using any other known means without loss of generality.

Next, processing logic uses the computed epipolar line to define a search range for finding matching pixels in the second image. In particular, at processing block 206, processing logic constrains the search range with respect to the epipolar line based on an indication of a desired correlation between efficient coding and semantic accuracy of matching. In particular, the desired correlation defines the proximity of the search range to the epipolar line (e.g., if the importance of semantic accuracy overweighs the importance of efficient coding, processing logic constrains the search range to be closer to the epipolar line, and vice versa). In one embodiment, the indication of the desired correlation between efficient coding and semantic accuracy is specified by a user via a user interface

provided by processing logic. In one embodiment, an initial seed is used to further constrain the search range with respect to a specific location on the epipolar line, as will be discussed in more detail below. In another embodiment, a seed is not used. Rather, the first and second frames are rectified to satisfy the property that for any pixel in a frame, the corresponding epipolar line in the other frame must be horizontal and scanline aligned with the location of the pixel. Rectification is the process of computing a homography, or 2D projective warp for each of the two frames, and applying these warps. Various algorithms can be used to compute the rectifying homographies and warp the frames accordingly.

[0031] At processing block 208, processing logic searches the second frame within the search range determined at processing block 208 for a match of the one or more pixels from the first frame.

[0032] Afterwards, at processing block 210, processing logic computes a motion vector based on the change of position of the pixels.

[0033] **Figure 3** is a flow diagram of one embodiment of a process 300 to define a search range for motion estimation using a seeding approach.

[0034] Referring to **Figure 3**, processing logic begins with receiving an indication of a desired correlation between efficient coding and semantic accuracy of matching from a user (processing block 302). In one embodiment, the indication is received by communicating to the user a user interface that allows the user to indicate the desired correlation between these two factors. For

example, the user interface may present the cumulative weight or percentage for both factors (e.g., 100%) and allow the user to distribute the cumulative weight or percentage between the two factors (e.g., the user may assign 20% to the efficient coding factor and 80% to the semantic accuracy factor). Alternatively, the user interface may provide a slider activated by the user to specify the desired correlation between the two factors.

[0035] At processing block 304, processing logic finds the position of an initial seed on the epipolar line for the block being coded. In one embodiment, the initial seed is found using a disparity vector. Disparity vectors are computed by stereo algorithms for each pixel of the first image to find the most semantically accurate match in another view that captures the same scene at the same snapshot in time but from a different viewpoint. When such disparity vectors are available, a relevant disparity vector is used as an initial seed.

[0036] At processing block 306, processing logic determines parameters of a window around the initial seed and the epipolar line based on the desired correlation between efficient coding and semantic accuracy of matching. In one embodiment, the metric for discerning the optimal match for the block being coded can be expressed as follows:

$$\vec{mv}_{best} = \underset{\vec{mv} \in S^{w}_{disp}}{\arg\min} [\, SAD(\vec{mv}) + \lambda_x . mvbits^{x}(mv_d^x) + \lambda_y . mvbits^{y}(mv_d^y) \,]$$

wherein mv_d^x, mv_d^y correspond to motion vector displacements from the disparity vector seeded location, respectively parallel and perpendicular to the epipolar

line. S_{disp}^{w} corresponds to the set of all candidate predictors in a search window (size $w \times w$) around the seed location provided by the disparity vector. *SAD* denotes the pixel-by-pixel sum of absolute differences between the block to be coded and the predictor block indicated by the candidate $\vec{mv}$. $mvbits(mv_d)$ represents the number of bits required to code the differential motion vector (it is an increasing function of the magnitude of the motion vector). λ is the appropriate Lagrange multiplier corresponding to the target quality.

[0037] By decreasing the relative values of λ_x with respect to λ_y, processing logic can force the match to lie closer to the epipolar line ensuring better semantic correctness. The case $\lambda_y = \infty$ corresponds to a search along the epipolar line. This special case involves just the use of the multi-view geometry to obtain the epipolar line obviating the need to obtain disparity vectors.

[0038] The use of disparity vector as a seed for motion search enables a relatively small search window around the seed and results in matches that have good compression efficiency and semantic accuracy. In addition, the use of a smaller search window leads to a reduction in complexity of the search process. Further, because good semantic matches are usually close to the epipolar line, the search becomes mostly 1-D (rather than a 2-D search in standard motion estimation), thus further reducing the complexity of the search process.

[0039] **Figure 4** illustrates two exemplary frames of a multi-view video sequence. For a point being coded in a frame 402, a disparity seed is found in a

previous frame 404. A window 406 illustrates conventional motion estimation, in which a search range defined around the disparity seed is not constrained based on the position of the epipolar line. A window 408 illustrates motion estimation that incorporates epipolar geometry constraints to influence the search range defined around the disparity seed and close to the epipolar line. A window 410 illustrates motion estimation that targets high semantic accuracy by limiting the search range to lie along the epipolar line.

[0040] **Figure 5** illustrates a comparison of bitrates obtained by experimenting with different motion estimation methods for a multi-view video sequence. The first column shows the bitrate for the case of standard motion estimation, where the motion search uses disparity vector as a seed for motion search. This method uses a small search window around the seed and provides for matches having good compression efficiency. This corresponds to the case when the relative weights of the horizontal and vertical Lagrange multipliers are similar, resulting in maximum compression.

[0041] Columns 2 and 3 show the case with a stronger bias towards keeping the match close to the epipolar line (greater horizontal Lagrange multiplier). These result in better semantic accuracy since the match is being constrained closer to the epipolar constraint, but the compression efficiency is reduced as a result.

[0042] The following description of **Figure 6** is intended to provide an overview of computer hardware and other operating components suitable for

implementing the invention, but is not intended to limit the applicable environments. **Figure 6** illustrates one embodiment of a computer system suitable for use as an encoding system 100 or just a motion estimator 104 of **Figure 1**.

[0043] The computer system 640 includes a processor 650, memory 655 and input/output capability 660 coupled to a system bus 665. The memory 655 is configured to store instructions which, when executed by the processor 650, perform the methods described herein. Input/output 660 also encompasses various types of computer-readable media, including any type of storage device that is accessible by the processor 650. One of skill in the art will immediately recognize that the term "computer-readable medium/media" further encompasses a carrier wave that encodes a data signal. It will also be appreciated that the system 640 is controlled by operating system software executing in memory 655. Input/output and related media 660 store the computer-executable instructions for the operating system and methods of the present invention. The motion estimator 104 shown in **Figure 1** may be a separate component coupled to the processor 650, or may be embodied in computer-executable instructions executed by the processor 650. In one embodiment, the computer system 640 may be part of, or coupled to, an ISP (Internet Service Provider) through input/output 660 to transmit or receive image data over the Internet. It is readily apparent that the present invention is not limited to Internet access and Internet web-based sites; directly coupled and private networks are also contemplated.

[0044] It will be appreciated that the computer system 640 is one example of many possible computer systems that have different architectures. A typical computer system will usually include at least a processor, memory, and a bus coupling the memory to the processor. One of skill in the art will immediately appreciate that the invention can be practiced with other computer system configurations, including multiprocessor systems, minicomputers, mainframe computers, and the like. The invention can also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network.

[0045] Various aspects of selecting optimal scale factors have been described. Although specific embodiments have been illustrated and described herein, it will be appreciated by those of ordinary skill in the art that any arrangement which is calculated to achieve the same purpose may be substituted for the specific embodiments shown. This application is intended to cover any adaptations or variations of the present invention.